IOP Conference Series: Materials Science and Engineering

Design and Construction of a Weather-Based Automatic Sliding Window

Design and Construction of a Weather-Based Automatic Sliding Window

Ademola Abdulkareem[1], T.E. Somefun[1*], V. Oguntosin[1], B.O. Adeyemi[1]

[1]Electrical and Information Engineering Department, Covenant University, Ota, Ogun State, Nigeria
Corresponding Author; tobi.shomefun@covenantuniversity.edu.ng, +2348037632337

Abstract-
In the twenty-first century, technology has continued to evolve rapidly and has found more applications in various sectors of life, especially in the control and monitoring of processes. The idea of automating appliances and control has been implemented in the automatic sliding window. Automation, indeed makes human living less stressful, which is the goal of engineering as a field. The automatic sliding window is designed with the primary aim of programming the action of the window per time to depend on environmental conditions utilizing sensors. The methodology, design, implementation, and testing of the system is explained in detail in this study.

Key words: technology, control, automation, sliding window

1. Introduction

Automation is the use of various control systems for operating equipment with minimal or reduced human intervention. It started in the late eighteenth century and became more popular in the nineteenth century. It has many advantages over the social or manual operation of any device, which accounts for the extensive research and exploration globally [1]. As technology is finding diverse applications in the control, automation, and monitoring of several devices, it has become necessary to implement these features in the design of doors and windows. The automatic control of windows has been worked on in a few projects. Still, it has not been widely integrated with the society, unlike the automatic doors which are in use in almost every location around the world. The concept of having a window started with making a small hole in the wall of a building and covering it with animal hide to prevent unwanted objects and harsh weather conditions. This was not effective as it could not stop the rain, and it was not safe for the inhabitants.

The windows' design has varied continuously in the shape, size, and materials used for its construction and the frame used to support it. Several materials have been used for windows, including papers, wood, vinyl, steel, aluminum, stained glass, and clear glass [2]. The foremost materials mentioned were used in the following period when the animal skin failed and gradually evolved to the latter elements. The frames were also designed using steel, fiberglass, aluminum, etc. but aluminum has proved to be a better choice majorly because it is relatively cheap and has resistance to corrosion. In this work, automation is applied to the window to control the opening and closing time concerning weather conditions.

1.1 Literature Review

IOP Conf. Series: Materials Science and Engineering 1107 (2021) 012179 doi:10.1088/1757-899X/1107/1/012179

This sub-section deals with the necessary theoretical framework and concepts required to understand the scope of this study, as was discussed in the introduction section. It seeks to delve into the procedures, techniques, and methods that have been employed previously to solve issues relating to this work. Different works and methods adopted would be studied and compared, and the shortcomings identified and assessed (Reference?).

1.1.1 History of windows

A window is an opening which passage to light and or air. It usually has an inner frame that allows for the opening or closing and a lock mechanism [3, 4]. It used to be made of opaque material but the modern windows are usually transparent. Windows have several functions: decoration to the building, security, thermal insulation, lighting, ventilation, and sound insulation, to mention a few [5]. Far back at 14th Century, windows are made with animal horns, but this didn't last as glassmakers started looking for ways to make windows out of glass. But then in ancient China, Korea, and Japan, paper window was dominant while the well-to-do Europeans preferred the mullioned window. These windows have vertical support that separates the tall narrow windows on both sides [6]. Glass windows were first adopted by the Romans. During early 17th Century, glass windows became affordable in England. At the end of the 17th century, however, the French developed several methods of producing larger plates of glass with better optical features [7]. Modern windows are usually filled with glass. There are several types of windows in use around the world today, namely, fixed windows, sliding windows, pivot windows, double-hung windows, single-hung windows, casement windows, awning windows, and skylight [8-10]. The configurations in which windows are made, and the closing and opening mechanisms vary with windows.

1.1.2 Related works

Authors in reference [11], a model was proposed for domestic use, which closes the window when raindrop is sensed. It was achieved with the use of rain sensors and a linear actuator. However, it was limited to environments where there is a constant downpour of rain, which makes it unsuitable for areas that had harsh weather conditions. Also, humanitarian aid is still required to open up the window when the rain had ceased to fall, which defeats the entire purpose of full automation. Authors in reference [12] worked on an automatic sliding window to overcome the problems faced with windows operated manually via remote control. The authors introduced a switch with a time relay sensor. When the button on the switch is pressed, current flows to the transformer, which causes the motor to start to move. The movement of the motor opens the window, and once it touches a limit switch at any end of the window, it stops its action there. The movement stops because the flow of current into the motor also stops. The window automatically closes because there is a time relay sensor and an IC in the circuit which enables this. The IC sends current to the sensor, which signals the motor to close the window in the opposite direction. On closing the window, a limit switch is reached, too, which stops further movement [12].

The work made the control of windows fully automatic and attractive for all classes of people in the society as it is safe and reliable for use. However, the major setback was the amount of energy consumed by the motor. This made the work inefficient in managing limited energy resources. Authors in [13] used sensors and an actuating hardware system to control of window's shade. The authors considered the winter period and summer period in their model. During the winter, heating is required, and the shade blinds were open to allow the maximum influx of light and vice versa in the summer. The curtains were designed to be open in the day time to a certain extent, but to be completely closed at night time no matter the weather condition. This was to allow for the privacy of the occupant [13]. In the author's design, the motor was connected to the bottom rail of the shades through a set of reduction gears, which allows for full and fast motion of the shadows. The motor was controlled using a personal computer and a digital data acquisition and control board. A button had to be pressed to vary from one mode to the other based on the season. Although the work was efficient in terms of energy usage, the control of blinds by occupants was eliminated. The major drawback of this work is that the window will be permanently closed during the winter. From the preceding, it is evident to improve upon previous work. However, this work seeks to improve on the limitations of past work and make the automatic control of windows to be as effective as possible.

1.2 System Design
This section explains in detail the processes involved in making the proposed design for functioning appropriately. It deals with both the hardware and software requirements that are involved in building the automatic sliding window. Also, the calculations required for the circuit and the working principle of some major components used are explained.

1.2.1 Hardware Components
Hardware components are substantial parts of the design and contain supply power, microcontroller, rain sensor, temperature sensor, H-bridge, LCD, and motor. Figure 1 shows the block diagram, while Figure 2 shows the circuit diagram. Some of the hardware components have sub-components attached to them, such as a resistor, diode, capacitor, etc. Specific functions of these subcomponents can be found in the open literature [14, 15].

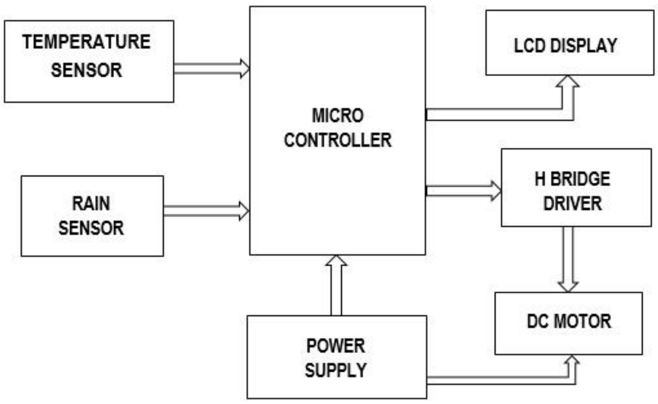

Figure 1: Block diagram for the automatic sliding window

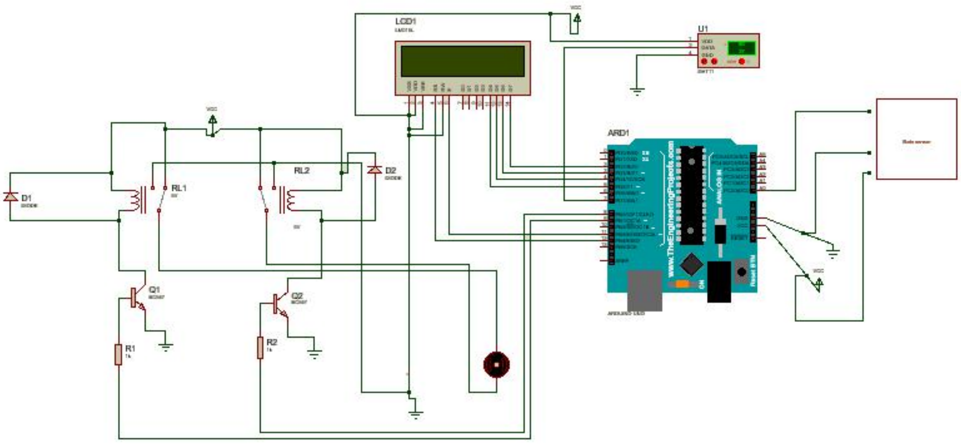

Figure 2: Schematic diagram of the automatic sliding window

a. Power supply

A dc voltage source is required to power different components with different voltage values such microcontroller (5 volts), DC motor ().

b. Microcontroller

The microcontroller has various family range. In this work, Arduino Uno found suitable; it is an open-source package. This microcontroller board is ATmega328P. It comprises 14 digital input/output pins, six analog inputs, a 16MHz quartz crystal, a USB connection, and a power jack In-Circuit Serial Programming header and a reset button. The 14 digital pins on the Uno operate at 5 volts, and by using different functions, they are set as either an input or output. Each pin receives between 20-50 mA which is recommended. It has an internal pull-up resistor of 20-50 kΩ. A maximum value of 40mA must not be exceeded on any I/O pin to guide against total damage [16]. The Arduino is shown in Figure 3

Figure 3: Arduino Uno

c. Liquid Crystal Display (LCD)

LCD is the technology used for display in digital watches, notebooks, and other small computers. There are various sizes of LCD, which are represented by an "m x n" notation. The m represents the number of columns, and n represents the number of rows. The "16 x 2" LCD indicates that there are 16 columns and two rows. All LCDs have two registers, namely, command and data registers [17, 18]. An

LCD is shown in Figure 4. LCD has 16 pins, the PINs, name, logic state, and description can be found in [19].

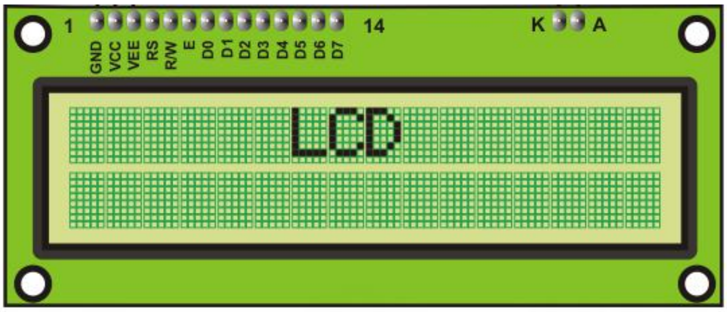

Figure 4: Liquid crystal display

d. DC motor

A DC motor is any class of rotary electrical machines that converts DC electrical energy into mechanical energy. The DC motor's working principle is based on the fact that when magnetic and electric fields interact, a mechanical force is produced. When a current-carrying conductor is placed in a magnetic field, it experiences a torque and tends to move. DC motors have no polarity, meaning that the two wires from the supply can be swapped to reverse the motor direction. Reversing the polarity of the applied voltage can be accomplished via the use of a motor driver capable of outputting both positive and negative voltage [20].

e. H-bridge driver

An H-bridge is an electronic circuit that aid application of a voltage across a load in either direction. An H-bridge is a simple circuit containing four (solid-state or mechanical) switching element, with the load at the center, in an H-like configuration [21, 22]. It is called H-bridge because the configuration of the switches and load are in a manner that resembles the alphabet 'H.' H Bridge drivers find application mostly in DC motors that run in both directions. The configuration of an H-bridge driver is shown in Figure 5.

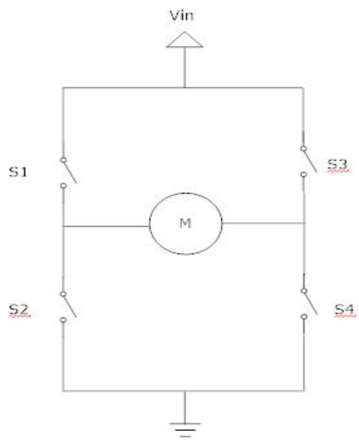

Figure 5: H-bridge driver

The primary operating mode of an H-bridge is simple: By closing switches S1 and S4, the right lead will be connected to the ground and current will start flowing through the left lead of the motor and energizes the motor to spin in the forward direction. The motor will spin to opposite direction when switches S2 and S3 are closed [22]. The switches S1 and S2 or S3 and S4 should never be closed simultaneously, because it will cause a short circuit on the input voltage source. In general, all four switching elements can be turned on and off independently, though there are some apparent restrictions [20]. Table 1 shows the various switching possibilities and the effect on a motor.

Table 1: Various switching configurations and the result on a motor

S1	S2	S3	S4	Result
1	1	0	0	The motor will stop (brake)
1	0	1	0	The circuit will burn
1	0	0	1	The motor will spin (forward)
0	1	1	0	The motor will spin (backward)
0	1	0	1	The circuit will burn
0	0	1	1	The motor will stop (brake)

The H-bridge has several models using different electrical components that act as switches. The model in use involves two relays, two NPN transistors, and resistors. The relays are Single Pole Double Throw (SPDT) relays, which have three contacts and two coils. The contact pins are normally closed, common, and normally open. A relay is either normally closed or normally open by default. For a normally closed relay, when power is received, it changes state. The common pin switches by completing a circuit with the normally closed pin and complete another loop with the normally open pin. This switching between the two states is what causes the motor to move in both directions. When one motor is actuated, and the other is not, the motor will run in one direction, and when the other motor is actuated, and the first is not, the reverse happens. The resistor limits the current coming into the transistor, and the value is calculated using Ohm's law. The transistor is the switching circuit, which switches the relay on when it has received sufficient current from the supply—a diagram showing how the coils and contacts are connected, shown in Figure 6.

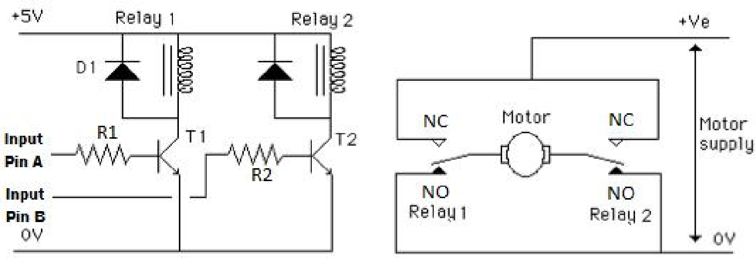

Figure 6: Modelled H-Bridge

A flywheel diode is connected in parallel with each relay coil, and since it is reverse biased when the relay is energized, it does not conduct. The diode serves as a conducting path in instances when the transistor is suddenly turned off. Since the coil of a relay is an inductor, it cannot change current instantly. When the motor is suddenly turned off, the circuit's energy has to follow the path provided by a diode. In the absence of this, a voltage transient will occur based on the equation (Ldi/dt), thereby causing an arc or damage. The numerator, di is finite (and negative), and the denominator, it very small, and since the current change is sudden (< 1μs), the voltage across the coil's inductance is going to be significant and negative. The danger is that the massive negative voltage spike may damage the transistor. However, the diode is forward biased by the spike and mostly shorts out the inductor and dissipates the spike [23, 24]

f. sensor

Two primary sensors used in this study, namely the rain sensor and temperature sensor, are used to detect rain and the degree of temperature in the environment, respectively. A rain sensor is an easy tool for rain detection; it detects water beyond what a humidity sensor can detect. The module features a rain board that collects the rain, a control board separate for more convenience, and a potentiometer for adjusting the sensitivity. On the raining board, there are two hook-up pins for polarity connected to two corresponding pins on the sensor board. The other pins on the control board represent the VCC, Ground, Analog output, A0, and Digital output, D0. It has a digital output LED illuminating when moisture has exceeded the threshold set by the sensitivity adjustment and power LED that indicates the power applied. The sensor board acts as a variable resistor that will change from resistance in Mega-ohms (MΩ) when dry to kilo-ohms (kΩ) when wet [25]. As a result, the more humid the board, the more current that will be conducted, and conversely, the lesser the water, the higher the output voltage on the analog pin. When connected to a 5V power supply, as long as no raindrop has been detected, the digital output is high, and the output LED will remain off [26]. When a little water is dropped on it, the digital output becomes low, and the output LED comes on. It is the digital output that is monitored, which gives an accurate indication of whether it is raining or not [27]. The rain board and sensor board are shown in Figure 7.

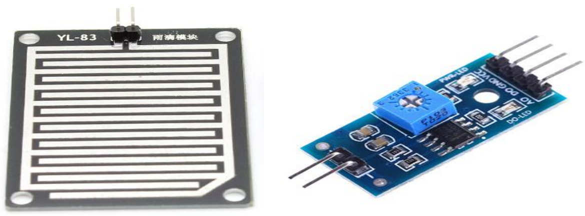

Figure 7: Rain sensor

The sensors connect to the Arduino, and based on received signals, the window either opens or closes. An H-bridge driver connected to the Arduino gets input from the microcontroller and gives an output that controls the window. An LCD displays the temperature, and humidity levels sensed in the environment per time. A particular limiting temperature of 36ºC is set, and when the temperature exceeds this limit, the window opens. As the temperature gradually reduces and becomes lower than 36ºC, the window closes back. However, if it begins to rain and the window is opened, the rain sensor overrides the temperature sensor, and the window closes. Temperature and humidity are inversely proportional; so, an increase in one parameter leads to a decrease in the other. As the window either opens or closes, the action per time is displayed on the LCD

2. Software Implementation

An Arduino can be programmed with a range of coding languages such as C, C+, and C++. The Arduino Uno is programmed via the Arduino Software Integrated Development Environment (IDE). The Arduino Uno in use is programmed with C language. Figure 8 presents the various processes involved in the acquisition of information to the control actions taken.

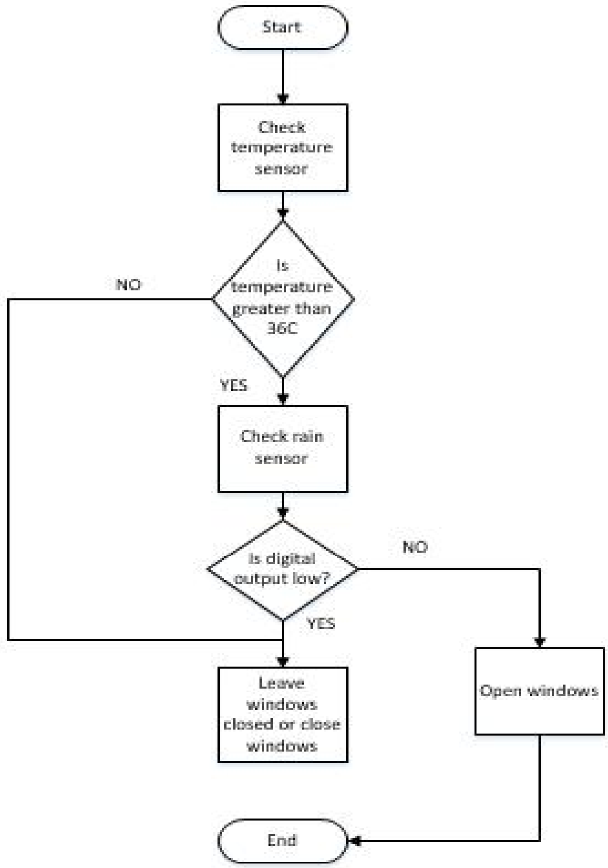

Figure 8: Flowchart showing the control process of an automatic sliding window

3. Implementation

The implementation of the automatic sliding window is divided into the outer framework and electrical aspect. These are discussed in details below.

3.1 Framework

The outer framework was constructed to resemble a window structure and an opening was left for the motion mechanism to be observed.

3.2 Electrical circuitry

The components were connected and tested to ensure adequate functionality before final package of the physical framework. The electrical components used were simulated on a Proteus software as shown in shown Figure 9.

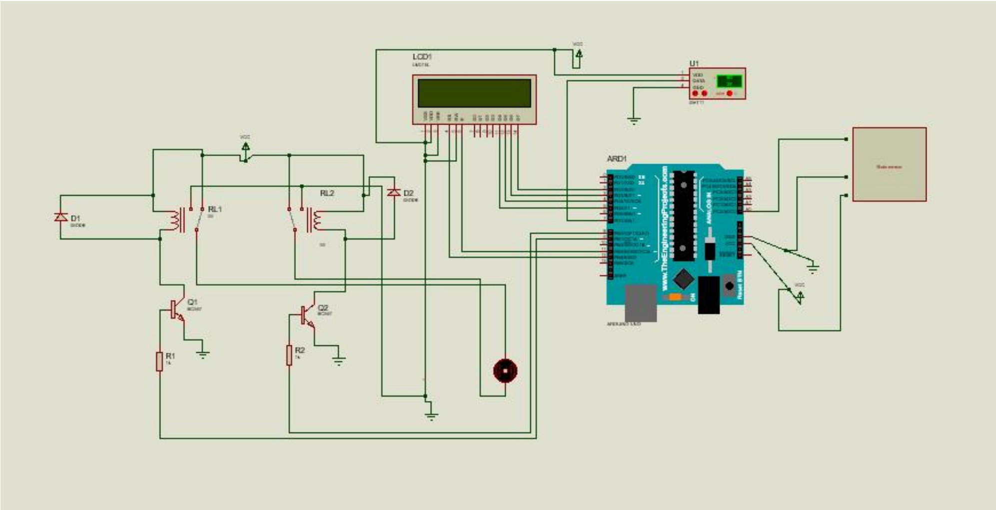

Figure 9: Simulation diagram of an automatic sliding window

3.3 Implementing the driving mechanism

The driving mechanism used was a gear. Gears were connected to the dc motor, and the motor determined the direction of motion of the window. When the motor moved in the forward direction, the gears also moved that way and when direction was reversed, it caused the gears to exhibit a corresponding motion. The drive mechanism is shown in Figure 10.

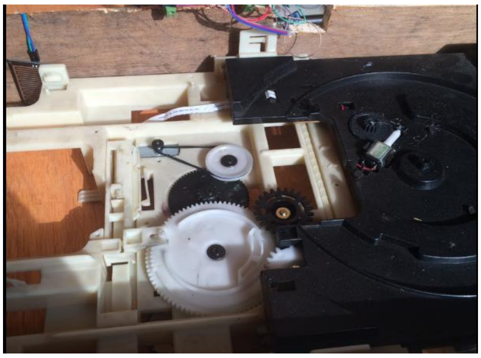

Figure 10: Driving mechanism of the automatic sliding window

3.4 Vero board and soldering

The Vero board is similar to bread board but with the major difference in how components are connected to the board. For a Vero board, all components have to be soldered to the surface and the boards can either be a continuous board or one without continuity. A continuous board has its continuity predetermined but a non-continuous board has its continuity determined by the individual using it. A non-continuous vero board was used and the components were soldered on it with the use of a soldering led as shown in Figure 11. At each point of soldering, the continuity was checked with the use of a multimeter and caution was taken to avoid bridging of components.

IOP Conf. Series: Materials Science and Engineering 1107 (2021) 012179 doi:10.1088/1757-899X/1107/1/012179

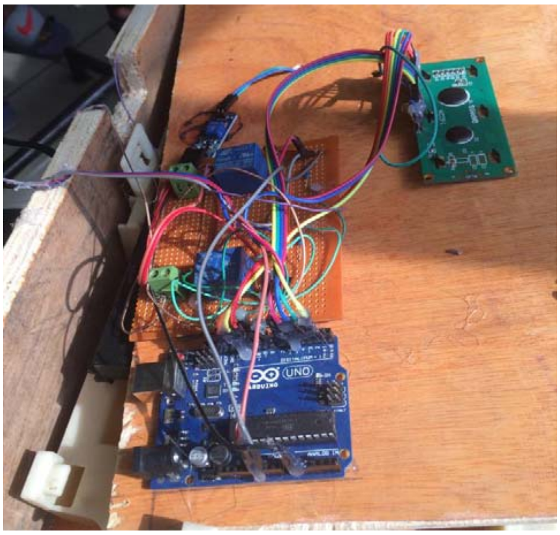

Figure 11: Patching of components on Vero board

3.5 Microcontroller programming

The microcontroller used was an Arduino Uno which was programmed with C language in an IDE (Integrated Development Environment). The microcontroller is like the central processing unit which responds to input signals from sensors in order to trigger a response to the component or device connected to the output. Based on the input signal received, the corresponding action to be carried out is determined in the code.

3.6 Defining variables

In this section, the variables used were initialized. The function used to achieve this was the *int* function which is used for storing the primary data type which are integers. On the UNO, a 16-bit value is stored which follows the syntax *int var = val* where var is the variable name and val is the assigned value to the variable. The sensors were declared using this function. The *const int* is a function used to imply that the data does not vary. An initial value is assigned to a variable that cannot be changed later in the program. The motor terminals were declared using this function shown in Figure 12.

```
int pino_a = A0; // pin connected to a0 sensor
int val_a = 0 ;
//const int pwm = 6;  //initializing pin 2 as pwm
const int in_1 = 8;
const int in_2 = 9;
```

Figure 12: Code for declaring variables

3.7 Libraries inclusion

The Arduino IDE has pre-installed libraries for special components but more can be downloaded if the required component's library is not available. For components that have a library, the components' libraries have to be used for the written code to function properly and this is done using the *include* function. The temperature sensor used which is DHT11 did not have a pre-installed library so it had to be downloaded before adding it in the code using the function. The LCD on the other hand, had a library that was just included at the beginning of the code as shown in Figure 13.

```
//const int vibrator =10;
#include <dht.h>
#include <LiquidCrystal.h>
```

Figure 13: Code showing the include function

3.8 Declaring variables

The functions used for this section were the *pinMode* and *void setup* function. The *void setup* function runs only once after each power-up or reset. The *void setup* function is used in declaring functions and this is where initial actions are performed. The *pinMode* is used to configure a specified pin to behave as specified; either as an input or output. The syntax it follows is *pinMode (pin, mode).* The 'pin' is the number of the pin to be set and the 'mode' varies between input and output. In this section, the motor terminals were declared as the output which receives input signal from the microcontroller as shown in Figure 14.

```
void setup(){
  //pinMode(pwm,OUTPUT) ;   //we have to set PWM pin as output
pinMode(in_1,OUTPUT) ;  //Logic pins are also set as output
pinMode(in_2,OUTPUT) ;
  lcd.begin(16, 2);
  // pinMode(vibrator, OUTPUT);
   pinMode (pino_a, INPUT);

}
```

Figure 14: Code showing the declaration of variables

3.9 Continuous operation

Some operations have to be performed continuously as the data does not remain constant. Such operations are continuous in nature and are written using the *void loop* function. This function makes the statements in it to run for an infinite duration. The void loop function has sub-functions in it which *are analogWrite*, *digitalWrite* and *delay* as shown in Figure 15.

```
void loop()
{
   val_a = analogRead (pino_a);
  //digitalWrite(in_1,HIGH) ;
//digitalWrite(in_2,HIGH) ;
delay(1000) ;
```

Figure 15: Code showing void loop function

An *analogWrite* function generates a square wave that can be varied in the function while the *digitalWrite* function generates its waveform manually. This is done by varying the values between a high (5V) and a low (0V). If the pin is configured as an output using the *pinMode* function, its voltage corresponds to 5V for high and 0V for low. Whereas, if it is configured as an input, *digitalWrite* will enable or disable the internal pull up resistor as shown in Figure 16.

```
  if(DHT.temperature >=36)
  {
    digitalWrite(in_1,HIGH) ;// open
digitalWrite(in_2,LOW) ;
//analogWrite(pwm,255) ;
delay(7000);
//digitalWrite(in_1,HIGH) ;
//digitalWrite(in_2,HIGH) ;
delay(1000) ;
    lcd.setCursor(0,1);
    lcd.print("window open ");
    delay(1000);
  }
  else
  {
    digitalWrite(in_1,LOW) ;//CLOSE
digitalWrite(in_2,HIGH) ;


  }
```

Figure 16: Code showing loop function

The loop function is shown in Figure 16. In this section, the temperature sensor read the environmental temperature constantly which determined whether the microcontroller sends the control to either close or open the window to the motor driver. When temperature exceeded 36ºC, one relay received a high input while the other remained low, which caused the motor to move in one direction. As temperature fell below 36ºC, the reverse happened which caused it to move in the other direction.

3.10 System test

The system was tested by using a mustimeter to measure the voltage at each point of the circuit to ensure the components get the required voltage. The continuity was also tested with the mustimeter to prevent and ensure there was no bridging of elements. The rain sensor was tested by immersing it in water i.e. imitating rain drops falling on it, and observing the increase in humidity on an LCD. The temperature sensor was placed near a heat source and the temperature increase was monitored with the LCD.

4. Conclusion

The design and construction of a simple model of an automatic sliding window are achieved in this study. The control unit in the design is the Arduino Uno microcontroller. The automatic control is achieved by coding the microcontroller in C language to respond to signals from other peripherals attached to it, such as rain and temperature sensors. A portable dc motor is also connected to obtain opening and closing operation of the window. For further work, the remote control can be implemented for purposes where the automation process has to be overridden. These situations vary from where privacy is needed ignoring the atmospheric conditions, the need to use the air conditioner, and so on. The remote will close the window or open it to suit the taste of the user.

Acknowledgements

The authors wish to acknowledge Covenant University management for her financial support.

Reference

[1] A. B. Brush, B. Lee, R. Mahajan, S. Agarwal, S. Saroiu, and C. Dixon, "Home automation in the wild: challenges and opportunities," in *proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 2115-2124.

[2] M. Tarantini, A. D. Loprieno, and P. L. Porta, "A life cycle approach to Green Public Procurement of building materials and elements: A case study on windows," *Energy,* vol. 36, pp. 2473-2482, 2011.

[3] D. A. Nolte and P. E. Braun, "Window lock," ed: Google Patents, 1991.

[4] A. Erdman, E. Nelson, J. Peterson, and J. Bowen, "TYPE AND DIMENSIONAL SYNTHESIS OF CASEMENT WINDOW MECHANISMS," *American Society of Mechanical Engineers (Paper),* 1980.

[5] L. Iannaccone, "Vice-principal research: A window on the building," *Education and Urban Society,* vol. 18, pp. 121-130, 1985.

[6] T.-H. Tsien, "Raw materials for old papermaking in China," *Journal of the American Oriental Society,* pp. 510-519, 1973.

[7] A. Bernardi, F. Becherini, G. Bassato, and M. Bellio, "Condensation on ancient stained glass windows and efficiency of protective glazing systems: two French case studies, Sainte-Chapelle (Paris) and Saint-Urbain Basilica (Troyes)," *Journal of Cultural Heritage,* vol. 7, pp. 71-78, 2006.

[8] M. K. Urbikain and J. Sala, "Analysis of different models to estimate energy savings related to windows in residential buildings," *Energy and Buildings,* vol. 41, pp. 687-695, 2009.

[9] K. Hassouneh, A. Alshboul, and A. Al-Salaymeh, "Influence of windows on the energy balance of apartment buildings in Amman," *Energy Conversion and Management,* vol. 51, pp. 1583-1591, 2010.

[10] S. Chaiyapinunt, B. Phueakphongsuriya, K. Mongkornsaksit, and N. Khomporn, "Performance rating of glass windows and glass windows with films in aspect of thermal comfort and heat transmission," *Energy and Buildings,* vol. 37, pp. 725-738, 2005.

[11] A. M. K. Mohd, Mohd, A., Nemer, S., "Automatic Closing Window," *unpublished graduation project,* 2011.

[12] S. P. Patil, J. R. Dhabuwala, and L. A. Patel, "Automatic Sliding Window," *International Journal of Science and Research, ISSN,* pp. 2319-7064.

[13] G. R. Luecke and J. Slaughter, "Design, development, and testing of an automated window shade controller," 1995.

[14] R. H. Wilkinson, "A method of generating functions of several variables using analog diode logic," *IEEE Transactions on Electronic Computers,* pp. 112-129, 1963.

[15] B. Ross and J. Litster, "Potential function and probability distribution of a nonequilibrium system: The ballast resistor," *Physical Review A,* vol. 15, p. 1246, 1977.

[16] D. F. Sendoya-Losada, P. T. Silva, and F. B. Marín, "REMOTE LABORATORIES USING THE TRAINING MODULEM2CI," 2006.

[17] D.-K. Yang and S.-T. Wu, *Fundamentals of liquid crystal devices*: John Wiley & Sons, 2014.

[18] H.-W. Chen, J.-H. Lee, B.-Y. Lin, S. Chen, and S.-T. Wu, "Liquid crystal display and organic light-emitting diode display: present status and future perspectives," *Light: Science & Applications,* vol. 7, pp. 17168-17168, 2018.

[19] C.-F. J. Kuo and T.-L. Su, "Optimization of injection molding processing parameters for LCD light-guide plates," *Journal of Materials Engineering and Performance,* vol. 16, pp. 539-548, 2007.

[20] N. D. Tan, J. Lee, M. R. Yazid, and W. A. F. W. Othman, "Mechatronic System: Automated Window Curtain using LDR," *International Journal of Engineering Creativity & Innovation,* vol. 1, pp. 1-7, 2019.

[21] Z. Zhu, J. Kim, and K. Abbassian, "H-bridge drive circuit for step motor control," ed: Google Patents, 2011.

[22] A. Siman-Tov, "Apparatus and method for current control in H-Bridge load drivers," ed: Google Patents, 2010.

[23] Y. Watanabe, M. Hirakawa, and K. Kimura, "DC/DC converter having transistor switches with flywheel diodes and program for controlling the transistor switches," ed: Google Patents, 2007.

[24] S. Miura, T. Makino, and F. Suzuki, "Load drive circuit using flywheel diode," ed: Google Patents, 2004.

[25] N. R. Lynam, "Rain sensor mount for use in a vehicle," ed: Google Patents, 2003.

[26] P. A. Reddy, G. S. Prudhvi, P. S. S. Reddy, and S. S. Ramesh, "Automatic rain sensing car wiper," 2018.

[27] S. Gunputh, A. P. Murdan, and V. Oree, "Design and implementation of a low-cost Arduino-based smart home system," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, 2017, pp. 1491-1495.